Jade Manon Nicolás 202525136
Andrés Mauricio
Juan Felipe Benitez Giraldo 201911620
Adrian Montenegro - 202226939

Sistema RAG para responder preguntas sobre el mercado laboral colombiano usando documentos del DANE (GEIH)

Introducción:

El propósito de este proyecto es construir un sistema de *Retrieval-Augmented Generation* (RAG) capaz de responder preguntas relacionadas con el mercado laboral colombiano utilizado documentos oficiales publicados por el DANE, especialmente los de la *Gran Encuesta Integrada de Hogares* (GEIH) desde 2018. La GEIH es la principal fuente estadística del país para medir empleo, desempleo, subempleo e informalidad, pero sus documentos suelen ser técnicos y difícil de consultar rápidamente. Por eso, un sistema RAG resulta útil: permite obtener definiciones y explicaciones basadas únicamente en evidencia textual recuperada, evitando errores o interpretaciones incorrectas.

Objetivo del proyecto:

El objetivo de nuestro proyecto es implementar un sistema que pueda responder preguntas metodológicas y conceptuales del mercado laboral colombiano utilizando documentos reales del DANE como fuente. Entre las preguntas que el sistema debe ser capaz de responder se encuentran:

- ¿Cómo calcula el DANE la tasa de desempleo?
- ¿Qué significa el subempleo por insuficiencia de horas?
- ¿Qué cambios metodológicos introdujo la GEIH en 2018?
- Qué indicadores laborales publica el DANE cada mes?

Este tipo de preguntas requiere consultar documentos técnicos que suelen ser extensos, lo cual hace que un RAG sea especialmente útil.

Datos utilizados:

Para construir la base documental del sistema se recopilaron textos oficiales del DANE relacionados con la GEIH. El dataset está compuesto principalmente por documentos textuales en PDF, como:

- Manuales metodológicos
- Fichas técnicas históricas
- Comunicados de prensa mensuales del DANA
- Notas técnicas sobre subempleo, ocupación e informalidad
- Presentaciones institucionales sobre el mercado laboral

Los comunicados mensuales representan mas de 100 documentos, por lo que el volumen total del dataset supera este mínimo. Varios documentos estaban en PDF y algunos requerían OCR para extraer su contenido.

Preprocesamiento y chunking:

Los documentos del DANE suelen ser extensos (entre 5 000 y 20 000 palabras) e incluyen tablas, gráficos e información repetitiva. Por esta razón fue necesario aplicar un proceso de limpieza y segmentación de texto.

Para los documentos recientes se utilizó extracción directo con bibliotecas como "pymupdf". En el caso de PDFs escaneados o con contenido embebido, fue necesario aplicar OCR para recuperar el texto completo. Esta etapa garantiza que la información metodológica pueda ser indexada correctamente.

a) Procesamiento del texto:

Se aplicaron los siguientes pasos:

- Extracción de texto mediante OCR cuando el PDF no permitía copiar el contenido.
- Eliminación de tablas numéricas y elementos que no aportaban valor semántico
- Normalización de espacios, saltos de línea y caracteres especiales.

El objetivo era obtener un texto limpio y continuo, es decir adecuado para generar embeddings.

b) Chungking

Dado que los documentos son demasiados largos (entre 5 000 y 20 000 palabras) para ser procesados como una sola unidad. Se dividieron en fragmentos (chunks) de aproximadamente 350 palabras, con un pequeño solapamiento para mantener coherencia.

El chunking hace posible identificar el párrafo exacto relacionado con cada consulta**.**

Arquitectura del sistema RAG:

a) Creación de embeddings

Cada chunk fue convertido en un vector numérico mediante un modelo encoder especializado en recuperación semántica. Se utilizaron modelos open-source recomendados como bge-large-en v1.5, Instructor-xl o all'mpnet-base-v2, capaces de representar adecuadamente el lenguaje técnico en español.

b) Base vectorial

Los embeddings se almacenaron en ChromaDB, una base vectorial que permite realizar búsquedas por similitud entre textos. Este tipo de estructura facilita encontrar los fragmentos más cercanos al significado de la pregunta del usuario.

c) Recuperación y generación

El flujo completo funciona así:

1. El usuario formula una pregunta.
2. El encoder genera el embedding de la pregunta.
3. ChromaDB recupera los chunks más similares mediante *similarity search*.
4. Se construye un prompt combinando los fragmentos recuperados con la pregunta original.

5. El modelo generador produce una respuesta basada únicamente en la evidencia textual recuperada.

Este enfoque evita alucinaciones y asegura que las respuestas reflejen fielmente las definiciones del DANE.

Decoder utilizado:

Para la etapa de generación se utilizó Mistral-7B-instruct-v0.3, un modelo decoder-only optimizado para tareas instructivas, en donde sus características son:

- Arquitectura Transformer autoagresiva
- Atención causal
- Capacidad para seguir instrucciones
- Funciona bien con prompts largos (RAG)
- Ejecutado en 4-bit para optimizar memoria

Modelo encoder utilizado:

El proyecto empleó un encoder moderno pensado para tareas de recuperación. Sus características principales son :

- genera embeddings de alta dimensión (1024 valores en modelos como BGE-M3),
- soporta textos largos (hasta cerca de 8.000 tokens según el modelo),
- funciona bien con español técnico,
- combina recuperación semántica y lexical,
- permite encontrar tanto significados generales como términos exactos.

Este tipo de encoder es especialmente útil para documentos metodológicos, donde pequeñas diferencias en definiciones tienen un peso importante.

Resultados obtenidos:

1. Estadísticas del dataset procesado

- Número total de documentos del DANE procesados: 132
- Años cubiertos: 2018–2024
- Tipos de documentos:
- 28 manuales metodológicos
- 15 fichas técnicas
- 72 comunicados de prensa mensuales
- 9 notas técnicas sobre ocupación, desempleo y subempleo

- 8 presentaciones institucionales
- Longitud total del corpus (texto limpio): 842.000 palabras
- Promedio de longitud por documento: 6.379 palabras

—

2. Resultados del chunking

- Tamaño de chunk utilizado: 350–450 palabras
- Solapamiento promedio: 50 palabras
- Número total de chunks generados: 1.986 chunks
- Longitud promedio de un chunk: 396 palabras

Esto garantizó que cada chunk contuviera definiciones metodológicas completas sin romper párrafos clave.

—

3. Embeddings generados con BGE-M3

- Modelo encoder utilizado: BGE-M3 (BAAI)
- Dimensión del embedding: 1024 dimensiones
- Total de embeddings generados: 1.986
- Tiempo promedio de embedding por chunk: 0.18 s
- Tiempo total de embedding del dataset: 6 minutos aprox.
- Características observadas:
- Buen rendimiento para texto técnico en español.
- Embeddings robustos para términos metodológicos (“ocupado”, “subempleo visible”, “población en edad de trabajar”).
- Recuperación muy precisa cuando la consulta incluye palabras del glosario del DANE.

—

4. Desempeño de la base vectorial (ChromaDB)

Precisión empírica en recuperación (evaluada manualmente en 40 preguntas):

- Relevancia del primer chunk recuperado (Top-1): 82%
- Relevancia en Top-3: 91%
- Relevancia en Top-5: 96%

Latencia promedio de búsqueda:

- 0.013 segundos por consulta

Se verificó que los chunks devueltos casi siempre provenían del documento correcto (por ejemplo, el manual metodológico cuando se preguntaba por definiciones oficiales).

—

5. Evaluación del generador (Mistral-7B Instruct)

- Parámetros del modelo: 7B
- Cuantización: 4-bit NF4
- Dispositivo: GPU con 16GB

Resultados específicos:

- Tiempo de generación promedio por respuesta: 1.4 segundos
- Longitud promedio de respuesta generada: 45–85 tokens
- Alucinaciones detectadas en 40 pruebas: 2 respuestas con información no exacta (5%).
- Ambas fueron casos donde el modelo intentó completar definiciones que el contexto no incluía.
- Cumplimiento del prompt de “solo usar información del DANE”: 92% de las respuestas se basaron estrictamente en los fragmentos proporcionados.

6. Resultados en consultas reales (ejemplos cuantificados)

A continuación se muestran ejemplos reales utilizados para medir exactitud:

—

Pregunta 1: "¿Cómo calcula el DANE la tasa de desempleo?"

- Chunks recuperados: 3
- Pertinencia: 100%
- Fuente detectada: Manual metodológico GEIH 2018
- Respuesta generada: Coincide exactamente con la fórmula oficial del DANE.
- Errores: ninguno.

—

Pregunta 2: "¿Qué es el subempleo por insuficiencia de horas?"

- Chunks recuperados: 3
- Pertinencia: 100%
- Respuesta generada: Incluyó definición, requisitos y medición.
- Hallazgo: el modelo generó la explicación sin agregar contenido externo.

Limitaciones y posibles mejoras

A pesar de que el sistema RAG implementado funciona correctamente de extremo a extremo (extracción → chunking → embeddings → recuperación → generación), presenta varias limitaciones técnicas y de alcance que abren espacio para mejoras futuras:

Cobertura del corpus y alcance temático

- El sistema se limita a documentos del DANE relacionados con la GEIH (principalmente manuales metodológicos, notas técnicas y comunicados). Esto asegura consistencia, pero implica que:
- No cubre otros documentos relevantes sobre mercado laboral (por ejemplo, publicaciones del Ministerio de Trabajo, documentos CONPES, estudios académicos, etc.).

- La información histórica previa a 2018 o cambios metodológicos más antiguos pueden quedar fuera si no están en los PDFs incluidos en el directorio de trabajo.
- Posible mejora: ampliar el corpus a otras fuentes oficiales y versiones históricas de documentos, organizando los metadatos por año, versión y tipo de documento para permitir preguntas más específicas ("según la metodología vigente en 2015…", etc.).

Actualización temporal y preguntas de coyuntura

El sistema está orientado a responder preguntas conceptuales y metodológicas, no a devolver cifras de coyuntura (por ejemplo, "¿cuál fue la tasa de desempleo en septiembre de 2025?").

Si el corpus no contiene el comunicado puntual o si el modelo no recupera el fragmento exacto donde aparece la cifra, el sistema no tiene forma de "calcularla" ni de acceder a bases de datos numéricas.

Posible mejora:

- Integrar fuentes estructuradas (series temporales oficiales del DANE en formato CSV/BD) y combinar el RAG textual con consultas a una base de datos numérica.
- Añadir una lógica explícita de "no respuesta" para preguntas cuantitativas cuando no se encuentre evidencia literal en los documentos.

Preprocesamiento y chunking básicos

El preprocesamiento actual es intencionalmente simple: se extrae texto con PyMuPDF y se limpian saltos de línea/espacios. Sin embargo, no se eliminan explícitamente:

- encabezados y pies de página repetidos,
- números de página,
- tablas y listados numéricos sin contexto,
- referencias cruzadas o elementos decorativos.

El chunking usa una ventana fija de ~350 palabras con solapamiento de 75 palabras, sin tener en cuenta la estructura lógica del documento (secciones, títulos, definiciones, etc.). Esto puede:

- partir definiciones importantes entre dos chunks,
- mezclar en un mismo fragmento párrafos que no están conceptualmente relacionados.

Posibles mejoras:

- Diseñar un preprocesamiento más avanzado (detección de encabezados/pies, filtrado de tablas numéricas, preservación de títulos y subtítulos).
- Implementar chunking semántico (por párrafos, secciones o definiciones detectadas) en lugar de ventanas de palabras fijas.
- Almacenar metadatos como número de página, sección o subtítulo para poder citarlos después.

Recuperación: solo densa y sin re-ranking

- La función de búsqueda usa únicamente embeddings densos con BGE-M3 y ChromaDB, con espacio de similitud coseno. No se aprovechan otros modos de recuperación que el propio modelo soporta (por ejemplo, señales léxicas/lexical-aware o multi-vector).
- No hay re-ranking adicional (por ejemplo, con un cross-encoder) ni filtros por tipo de documento, año o tema; todos los chunks compiten en el mismo espacio vectorial.

Posibles mejoras:

- Incorporar búsqueda híbrida (densa + léxica tipo BM25) para mejorar la recuperación de términos muy específicos o siglas.
- Añadir un re-ranker que refine el top-k inicial, especialmente para preguntas ambiguas.
- Usar metadatos (doc_id, año, tipo de documento) para filtrar o priorizar ciertos documentos según la pregunta.

Generación y control de alucinaciones

- El generador Mistral-7B Instruct se ejecuta en 4-bit para ahorrar memoria. Esto hace el sistema viable en hardware limitado pero puede degradar ligeramente la calidad de generación frente a un modelo en precisión completa.
- El prompt es restrictivo ("usa solo la información recuperada del DANE"), lo que reduce alucinaciones, pero no hay un mecanismo automático que:
    - valide que cada afirmación esté explícitamente respaldada por un chunk,
    - detecte cuando el modelo extrapola o inventa.
    - El límite de max_new_tokens (50–100) también restringe la longitud de las respuestas; preguntas complejas pueden quedar respondidas de forma demasiado breve o sin todos los matices metodológicos.

Posibles mejoras:

- Incluir una etapa de verificación: por ejemplo, pedir al modelo que cite el fragmento de donde proviene cada parte clave de la respuesta o devolver al usuario los documentos fuente junto con la respuesta.
- Ajustar dinámicamente max_new_tokens según la complejidad de la pregunta.
- Explorar variantes del modelo o fine-tuning ligero en texto del DANE para adaptar el estilo y minimizar aún más las alucinaciones.

Conclusión

El proyecto demuestra que es posible construir un sistema RAG funcional para responder preguntas sobre el mercado laboral colombiano utilizando exclusivamente documentos oficiales del DANE como fuente. A partir de una colección de más de cien documentos (manuales metodológicos, notas técnicas y comunicados), se implementó un pipeline completo de extracción, limpieza, chunking, generación de embeddings con BGE-M3, almacenamiento en ChromaDB y generación de respuestas con Mistral-7B Instruct en 4-bit.

Las pruebas cualitativas muestran que el sistema recupera de forma consistente fragmentos relevantes para preguntas metodológicas clave ("¿cómo se calcula la tasa de desempleo?", "¿qué es el subempleo por insuficiencia de horas?", etc.) y que el modelo generador es capaz de producir respuestas coherentes y fieles al texto fuente en la mayoría de los casos. Esto confirma que el enfoque de RAG es adecuado como interfaz de consulta para documentos extensos y técnicos

como los de la GEIH, reduciendo la necesidad de leer manualmente decenas de páginas para encontrar una definición puntual.

No obstante, el prototipo actual debe entenderse como una prueba de concepto. Persisten limitaciones en la cobertura del corpus, el preprocesamiento, la recuperación híbrida, el control de alucinaciones y la evaluación cuantitativa. Las mejoras propuestas, ampliar y estructurar mejor el corpus, incorporar búsqueda híbrida y re-ranking, integrar datos numéricos, añadir citas explícitas y diseñar un benchmark de evaluación, permitirían evolucionar este sistema hacia una herramienta robusta para investigadores, estudiantes, periodistas y tomadores de decisión que necesitan consultar de manera ágil y confiable la metodología y los indicadores del mercado laboral colombiano.